



Experimental bifurcation analysis—Continuation for noise-contaminated zero problems

Schilder, Frank; Bureau, Emil; Santos, Ilmar Ferreira; Thomsen, Jon Juel; Starke, Jens

Published in:
Journal of Sound and Vibration

Link to article, DOI:
[10.1016/j.jsv.2015.08.008](https://doi.org/10.1016/j.jsv.2015.08.008)

Publication date:
2015

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Schilder, F., Bureau, E., Santos, I. F., Thomsen, J. J., & Starke, J. (2015). Experimental bifurcation analysis—Continuation for noise-contaminated zero problems. *Journal of Sound and Vibration*, 358, 251-266. <https://doi.org/10.1016/j.jsv.2015.08.008>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Experimental Bifurcation Analysis – Continuation for Noise-Contaminated Zero Problems

Frank Schilder^{*†} Emil Bureau[‡] Ilmar Ferreira Santos[‡]

Jon Juel Thomsen[‡] Jens Starke[†]

23rd June 2015

Abstract

Noise contaminated zero problems involve functions that cannot be evaluated directly, but only indirectly via observations. In addition, such observations are affected by a non-deterministic observation error (noise). We investigate the application of numerical bifurcation analysis for studying the solution set of such noise contaminated zero problems, which is highly relevant in the context of equation-free analysis (coarse grained analysis) and bifurcation analysis in experiments, and develop specialized algorithms to address challenges that arise due to the presence of noise. As a working example, we demonstrate and test our algorithms on a mechanical non-linear oscillator experiment using control based continuation, which we used as a main application and test case for development of the the COCO compatible MATLAB toolbox CONTINEX that implements our algorithms.

1 Introduction

In [1, 2] we have implemented and tested bifurcation analysis directly in a non-linear mechanical oscillator experiment using control based continuation [3, 4, 5, 6], that is, tracking stable and unstable responses under variations of forcing frequency and amplitude. In these experiments we used a harmonically excited impact oscillator with control forces applied via electromagnetic actuators as a test specimen and developed methods for tuning the control parameters and for determining stability of steady-state responses. Here, we will provide information about the details of the implementation of the toolbox CONTINEX developed for this type of experiment as well as for equation-free analysis [7, 8, 9, 10, 11]. We will motivate the algorithmic developments with new experimental results collected with the set-up used in [1, 2]. The presented methods are able to extract bifurcation diagrams directly from strongly non-linear experiments, and the toolbox CONTINEX is freely available for download [12].

Control based continuation in experiments [3, 4, 5, 6] and continuation for equation-free problems [7, 8, 9, 10, 11] are two examples of continuation problems that involve *noise contaminated zero problems* $F(u) = 0$, a term that will be made more precise in Section 2. In the context of control based continuation, which we discuss with an example in Section 3, the function F is commonly defined as the average of a control signal or the difference between an observed state and

^{*}Corresponding author: frans@dtu.dk.

[†]Department of Applied Mathematics and Computer Science, Technical University of Denmark.

[‡]Department of Mechanical Engineering, Technical University of Denmark

a target state. Here, typical sources of noise contamination are the noise level of measurements or the excitation of unobservable modes.

Equation-free methods allow to analyze numerically the macroscopic behavior of microscopically defined models without having an explicitly given equation for the macroscopic dynamics. In these multi-scale systems, where the macroscopic behavior is due to fast convergence of many degrees of freedom to an attractive low-dimensional slow manifold, the noise contamination is due to remaining fast scales when, in the observed time, the system behavior did not converge close enough to the slow manifold. The microscopic models could be even stochastic and intermediate system sizes show remaining noise also on the macroscopic level which leads to similar problems.

Both constructions of zero problems have in common that the result is a function F with the property that repeated evaluations with the same argument will produce different results. In other words, an evaluation of F at a point u can be interpreted as drawing a point from some unknown distribution. The goal of this work is a robust algorithm for computing solution branches of such noise contaminated zero problems, which allows running batches of continuations with a high rate of success and without supervision. We give an account of our experience collected during our development effort and of key ideas for tackling challenges we encountered in Section 4. The detailed technical description of our algorithms is subject of Section 5. We use data collected during runs with our non-linear mechanical oscillator experiment to illustrate important aspects of the challenges and proposed algorithms.

2 Noise-contaminated zero problems

Let us start by recalling the basic terminology of continuation as used in the context of toolbox development based on COCO [13, 12]. A *continuation problem* is the problem of computing an approximation to the solution manifold of a so-called *zero problem*. More precisely, let $F : \mathbb{R}^m \rightarrow \mathbb{R}^n$, $m \geq n + 1$, be two times differentiable with Lipschitz-continuous second derivative. Under the assumptions that $F(u^*) = 0$ for some $u^* \in \mathbb{R}^m$ and that the Jacobian $(\partial F / \partial u)(u^*)$ of F at u^* has full rank, there exists an $(m - n)$ -dimensional differentiable manifold $\mathcal{M} \subset \mathbb{R}^m$ of solutions through u^* and embedded in \mathbb{R}^m . We will restrict to the case $m = n + 1$ of most common interest in applications, that is, to solution manifolds of dimension 1, which conventionally are referred to as *solution branches* or simply as *branches*.

The task of a continuation algorithm is then to produce a so-called *bifurcation diagram* given an approximate initial point $u^* \in \mathbb{R}^m$ close to a branch. A bifurcation diagram of a zero problem is a collection of branches restricted to a computational domain and possibly connected in so-called *singular points* or *branch points*, which are points $u \in \mathbb{R}^m$ where the Jacobian $(\partial F / \partial u)(u)$ does *not* have full rank. A good continuation method will typically succeed in computing a connected component of a bifurcation diagram, as well as locate all singular points on this connected component. As a slight generalization, practically all continuation packages today [14] allow the computation of bifurcation diagrams involving different zero problems and, hence, branches representing different types of solutions. An example is a bifurcation diagram comprising a branch of periodic solutions of an ODE connected to a branch of equilibrium points at a Hopf-bifurcation point.

An algorithm for computing branches is the so-called method of *pseudo arc-length continuation* [15, 16]. This is a predictor-corrector method that is based on augmenting the zero problem $F : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ with the so-called *pseudo arc-length condition* $g(u) := t^T(u - u_0) - h = 0$, where t is a unit vector tangent to the branch at u_0 , u_0 is a point on the branch, and h is the continuation step-size. For known u_0

a new point u^* on the branch can be computed by applying some non-linear solver, the *corrector*, to the non-linear system of equations $\{F(u) = 0, g(u) = 0\}$, where the initial guess $u_1 = u_0 + ht$ provided as a starting point to the solver is called the *tangent predictor*. Geometrically, the pseudo arc-length condition defines a hyperplane through u_1 perpendicular to t , and the corrector locates the point of intersection of this hyperplane with the branch, which is approximately distance h away from u_0 .

In the context of COCO [13, 12], an algorithm for computing a branch is referred to as a *covering algorithm*; see also Chapters 1 and 3 in [16]. Covering algorithms follow the concept of predictor-corrector methods and pseudo arc-length continuation is an example of a covering algorithm for differentiable branches. On the level of abstraction introduced by COCO [13, 12], a covering algorithm is composed of three components: a corrector, a curve-segment and an atlas. An atlas is an instance of an atlas class, which implements at least methods for the execution of a predictor-corrector step, step-size control and handling of failure of convergence of the corrector. In addition, an atlas maintains a representation of a branch that is sufficient for the covering algorithm to proceed in a meaningful way. Similarly, a curve-segment provides for a local representation of a branch and a so-called *projection condition*. The pseudo arc-length condition stated above is an example of such a projection condition. Finally, the corrector implements a non-linear solver, for example, Newton's method. The toolbox COCO [13, 12] includes a generic driver that assembles a covering algorithm and executes the computation of a branch given a corrector class, a curve-segment class and an atlas class.

In this terminology, our aim is the development of a corrector, curve-segment and atlas class suitable for solution branches of noise contaminated zero problems. In order to motivate the requirements on such classes, we will state an intuitive definition of a branch together with some basic observations of fundamental differences to branches of smooth zero problems. Consider a zero problem $G(u) = 0$, $G : \mathbb{R}^m \rightarrow \mathbb{R}^n$, that satisfies the same smoothness conditions as stated at the beginning of this section. Assume further that this zero problem is not known explicitly, but can be evaluated only implicitly through observations $F(u) := G(u) + \varepsilon(u)$ that are subject to observation errors $\varepsilon(u)$. We call $F(u) = 0$ the *noise contaminated zero problem*, while we refer to $G(u) = 0$ as the underlying *true zero problem*. As an example, a discussion of such noise contaminated zero problems in the context of equation-free analysis can be found in [17].

In this setting, an intuitive definition of a solution branch is a set of points $U \subset \mathbb{R}^m$ for which the mean value $\mathcal{M}[F(u)]$ of the residuum is zero. In fact, this definition seems to be the only practicable one unless more detailed information about the observation error is available. Although this definition seems simple enough, there are important consequences. An immediate difference to a smooth zero problem is, that a repeated evaluation of F with the same argument will produce different values for each evaluation. A somewhat more subtle implication is, that the solutions of the equation $\mathcal{M}[F(u)] = 0$ will, in general, not coincide with the solutions of $G(u) = 0$. If the distribution of the observation error has mean different from zero, we can only expect that solutions of $\mathcal{M}[F(u)] = 0$ are close to solutions of $G(u) = 0$ if the Jacobian of G is sufficiently well conditioned. In particular, close to singular points of G we will have to expect significant deviations between solution branches of the two zero problems.

Even in the special case that the distribution of the observation error has mean zero we have to expect deviations, because in practice we can solve the equation $\mathcal{M}[F(u)] = 0$ with limited accuracy only. Any algorithm will be restricted to drawing a finite sample of observations, for which the empirical mean value of the sample is only an estimate for the actual mean. For example, if the observation error is normally distributed with mean 0 and standard deviation (SD) σ , the empirical

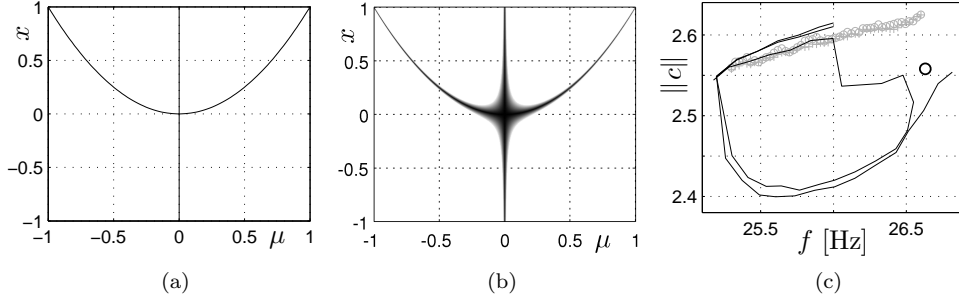


Figure 1: Effect of noise contamination on a zero problem. As an example, the solution set of the zero problem $\mu(x - \mu^2) = 0$ is shown in (a). It consists of four branches that meet in a branch-point at the origin. Panel (b) shows the solution set of $|\mu(x - \mu^2)| \leq 0.01$, illustrating the potential effect of a small perturbation ε onto the location of a zero of the perturbed zero problem $\mu(x - \mu^2) + \varepsilon = 0$. Panel (c) shows an observation from the experiment described in Section 3, which resembles the situation sketched in (b). The suspected location of a potentially existing singular point is marked with \circ . Due to significant increase of the standard deviation (SD) near a singular point, a continuation approaching the singular point in (b) from below would proceed with high probability either on the left-hand or right-hand branch, which is a likely explanation for the observation in (c). This is in stark contrast to a smooth zero problem (a), for which a continuation method would typically pass straight through the singular point and proceed on the middle branch.

mean value is normally distributed with mean 0 and SD σ/\sqrt{N} , where N is the size of the sample. In other words, to double the accuracy one has to quadruple the size of the sample. As a consequence, one will obtain branches close to branches of the true zero problem under the condition that the true zero problem is sufficiently well-conditioned. In addition, close to singular points one will typically observe a strong influence of noise; see Figure 1.

3 A mechanical non-linear oscillator experiment

As a test specimen for control based continuation we designed a mechanical non-linear oscillator experiment with a number of objectives in mind. The experiment should be simple to set-up, preferably requiring only standard equipment as used in student projects; it should have an electromagnetic actuation system that can be considered a prototype of active magnetic bearings in rotating machinery; and it should be dynamically more complex than a forced one degree of freedom system in order to extend experience of applying control based continuation beyond earlier studies using this method [3, 4, 5, 6].

Our experimental setup is shown in Figure 2(a) and is composed of two subsystems, a platform driven by an electromagnetic shaker and an oscillator mounted to the platform 2(b). In addition, two electromagnetic actuators are mounted to both sides of the tip mass of the oscillator, which allow to apply a control force directly to the tip mass. Both subsystems can oscillate in one direction only and, therefore, introduce one degree of freedom each. The coupling and mass ratios of both subsystems are such that the setup resembles a non-linear vibration absorber mounted to a structure that is subject to some external excitation. A stiffening spring non-linearity is realized by limiting the oscillation amplitude of a point of the beam that is part of the oscillator 2(c).

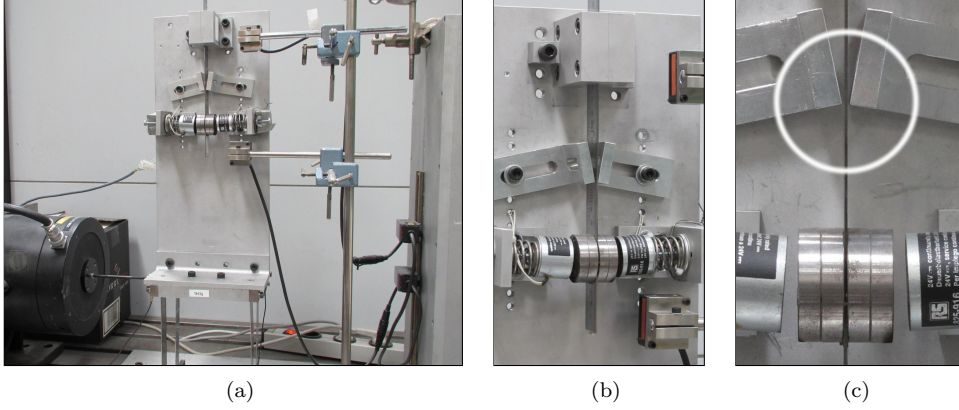


Figure 2: A mechanical non-linear oscillator experiment (a) consisting of a platform driven by an electromagnetic shaker and an oscillator mounted to the platform (b). The displacements of platform and oscillator are measured with laser displacement sensors installed to the right of the platform. Electromagnetic actuators mounted to both sides of the mass allow the direct application of a control force. A hardening spring non-linearity is realized by hard limiting the displacement of a point of the beam (c).

To construct a zero problem for exploring branches of periodic responses we follow the idea of control-based continuation [3, 4, 5, 6]. The key component of this method is a so-called non-invasive feedback control, which is realized in our experiment via a PD controller with input $x(t) - y(c, t)$ and control target $x(t) - y(c, t) \rightarrow 0$. Here, $x(t)$ is the response of the oscillator at time t , computed as the difference of the displacement measurements at the platform and tip mass, and

$$y(c, t) := c_1 + \sum_{k=1}^5 c_{k+1} \sin(k2\pi ft) + \sum_{k=1}^5 c_{k+6} \cos(k2\pi ft) \quad (1)$$

is a reference signal computed as a truncated Fourier expansion of order 5. The control signal generated by the controller is fed into the electromagnetic actuators after splitting and amplification; please refer to [1] for details. Ideally, for every chosen excitation amplitude A , frequency f and input vector $c \in \mathbb{R}^{11}$ of Fourier modes, the controlled experiment settles onto a globally unique response independent of the initial state. Although this requirement is unnecessarily strong and control-based continuation works well with control systems that are not globally unique in that sense, we will assume that it holds to simplify the present discussion.

A zero problem is now constructed as follows. Assume that $X(A, f, c)$ is a sufficiently large discrete sample of the response that the controlled experiment settles onto for the input A , f and c , and that \mathcal{F} is a function computing the first 11 Fourier modes of such a sample as in (1). Then,

$$F(c, A, f) := \mathcal{F}(X(A, f, c)) - c \quad (2)$$

defines a zero problem with the property that, for given A and f , every solution vector c of the equation $F(c, A, f) = 0$ provides an order-5 Fourier approximation of a natural state of the *uncontrolled* experiment, because for $F(c, A, f) = 0$ the input $x(t) - y(c, t)$ to the PD controller satisfies $\mathcal{F}(x(t) - y(c, t)) = \mathcal{F}(x(t)) - c \approx 0$. In other words, the control force applied via the electromagnetic actuators is essentially zero plus higher-frequency measurement noise and, hence, the function

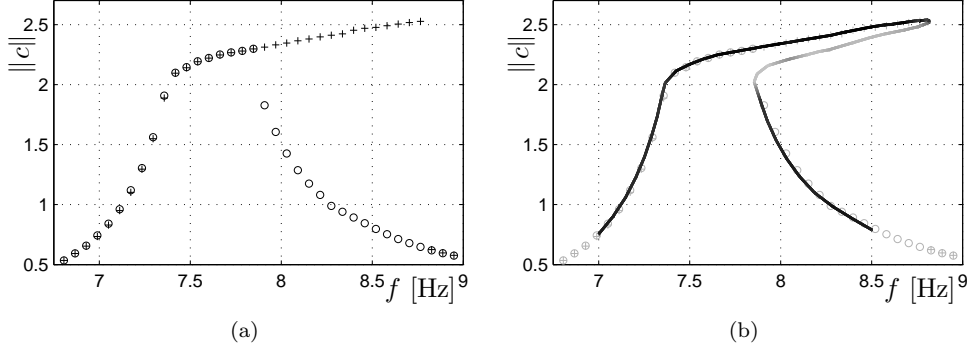


Figure 3: Frequency response diagrams of our mechanical non-linear oscillator experiment obtained from up- '+' and down- 'o' sweeps (a), and from a continuation including quantification of stability (b). For comparison, the results of the continuation in (b) are overlaid on top of the sweeps (gray). The value $\|c\|$ corresponds approximately to the response amplitude in millimeters. We observe a Duffing-oscillator type response diagram typical for a spring-mass-damper system with a hardening spring.

$y(c, t)$ provides an accurate approximation of a natural periodic response of the uncontrolled experiment.

Figure 3 shows a bifurcation diagram illustrating the dynamics of the oscillator under variations of the forcing frequency. We observe the typical response curve of a spring-mass-damper system with stiffening spring, exhibiting a large window of hysteresis and multi-stability. The bifurcation diagram in Figure 3(b) was obtained with the most recent version of our toolbox, which allows assessing stability of a response [2]. Stability is indicated in gray-scale along the response curve, black corresponds to asymptotically stable responses, while gray indicates instability. Extensive studies of the dynamics of this experiment and of a fitted one-degree of freedom model of the oscillator were conducted in [1, 2, 18].

Although our toolbox was developed having all types of problems in mind, which can be considered as noise contaminated zero problems, the remainder of this text focuses exclusively on observations we collected during control-based continuation applied to our mechanical non-linear oscillator experiment. The reason is a fundamental difference between an actual experiment and a computer simulation. While in any form of computer simulation, for example, during a run of a fine-grained model as part of an equation-free analysis, it is at least in principle possible to access every state variable explicitly, this is not true in experiments, where only limited information is available through measurements. This implies that in experiments there can be relevant states or modes that are not observable with a given setup of sensors. For example, in our experiment we are certain that for some forcing frequencies torsional modes of the beam are excited. However, these are not observable directly with our setup of displacement sensors, but nevertheless affect a control based continuation, because energy is transferred between observable and unobservable modes leading to additional and possibly systematic observation errors. Therefore, we consider control-based continuation applied to an actual experiment as the hardest test case for algorithm development. This working hypothesis is supported by our observation that all algorithms we tested during the course of our development that succeeded in the experiment also succeeded in simulations while the converse did not hold.

We started the development of the toolbox CONTINEX [12, 19] using basic atlas and curve segment tutorial classes as described in [13]. As a starting point for a cor-

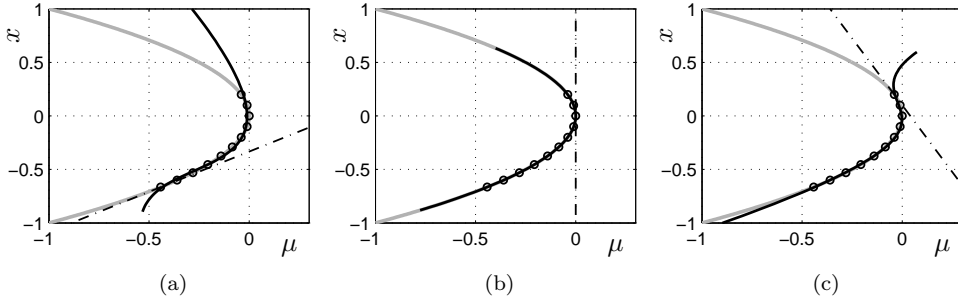


Figure 4: Approximation of a set of points, equally spaced with respect to arc-length along the parabola $\mu = -x^2$, using the algorithm of Section 4.1 and fitting to a third-order polynomial. Using the vertical tangent at the origin reproduces the parabola exactly (b), while other choices (here either end point of the sequence) result in non-zero third-order contributions (a) and (c). Note that in all three cases the approximations are close to the sample points, that is, the approximation polynomials *interpolate* with high accuracy. However, as seen in (a) and (c) the accuracy of an *extrapolation* of the solution curve is often observed to be rather poor.

rector we used a copy of a class implementing a Newton-like method with Broyden updates, which was shipped with COCO [13, 12]. In a first version we implemented algorithms as were used in early applications of control-based continuation [3, 4, 6].

4 A history of challenges

The toolbox CONTINEX [12, 19] was subject to significant evolution during the course of its development and, here, we will give an account of the major obstacles we encountered along the way, and of ideas how to overcome these. The contents of this section is based on notes taken during supervised experiments, which are hand-written and included as a scan in the data package [19]. The observations recollected below provide important background information motivating a number of non-trivial features of our algorithms. However, the formal description of our algorithms in Section 5 is kept independent of the observational material presented here.

4.1 Failure of polynomial approximations

In a first version of our toolbox we implemented a polynomial least-squares approximation of a curve segment similar to earlier work [3, 4, 6]. The objective is to construct a smoothened local representation of a branch to enable the application of an existing covering algorithm for differentiable branches, for example, pseudo arc-length continuation, which seems to be a natural starting point for development.

If we denote the normalized vector of the continuation direction with t and a sample of points with u_k , $k = 1, \dots, N$, and compute the base points $x_k := t^T(u_k - u_1)$ and normal deviations $y_k := u_k - u_1 - x_k t$, we obtain an overdetermined system of linear equations for computing the coefficients of an approximating polynomial $y = p_r(x)$ of degree $r < N$. The basic idea of continuation is to shift an approximation window in the direction of t along the branch with shift-size equal to the continuation step-size. Depending on the window size, consecutive windows can be disjoint or overlap.

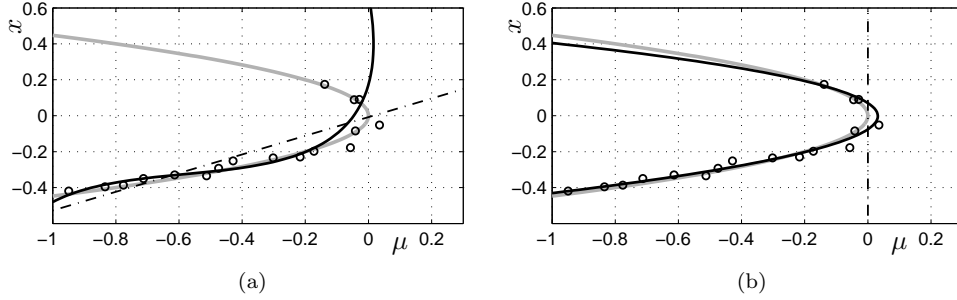


Figure 5: Approximation of a sequence of points, drawn from normal distributions with SD 0.025 and mean values equally spaced with respect to arc-length along the parabola $\mu = -5x^2$, using the algorithm of Section 4.1 and fitting to a third-order polynomial. Using the mean direction vector results in a polynomial that interpolates the sample reasonably well, but extrapolates poorly (a). Using the *a-priori* knowledge about the distribution of points along the parabola, that is, the vertical tangent through the origin, results in an approximation polynomial with significantly better extrapolation properties (b).

Although this approach seems feasible, we discovered serious problems, in particular, when passing through a sharp turning point. As it turns out, the quality of the polynomial approximation is quite sensitive to the choice of continuation direction; see Figures 4 and 5. Furthermore, fast uniform convergence of the approximating polynomials for increasing degree r relies on fast overall decay of the derivatives of the branch represented as a function of x within the approximation window. The latter condition can be satisfied, for example, if t is a vector tangent to some point of the branch inside the approximation window and if the size of the approximation window can be reduced arbitrarily. Unfortunately, to obtain a useful estimate for a tangent direction for noise contaminated zero problems, the size of the approximation window must be significantly larger than the SD of the observation error, and, hence, cannot be reduced arbitrarily. As a consequence, we observed regular and systematic failure of our continuation algorithm close to points where the branch has high curvature or where the SD of the observation error increases for some reason.

One could argue that some of the detrimental observations mentioned above are due to the fact that we use a simplified approximation problem with base points x_k that are computed *a-priori*. In the more general setting of non-linear total least squares, also referred to as curve approximation, the base points x_k are part of the unknowns of $a = \operatorname{argmin}_{\{a \in \mathbb{R}^{r+1}\}} \min_{\{x_k \in \mathbb{R}^m\}} \|p_r(x_k) - u_k\|_2^2$, where a is the vector of parameters of p_r . This least-squares problem is inherently non-linear for $r > 1$ and, therefore, would require an additional iterative solver. While using curve approximation is likely to result in better approximations as indicated in Figure 5, at this point we decided to follow the path of reducing the overall complexity of our toolbox by seeking simpler algorithms instead of implementing more complicated ones. As a result of this strategy, we arrived at an algorithm that uses a secant approximation through points sampled and averaged inside the corrector instead of along a curve segment, as outlined in Sections 5.2 and 5.3.

4.2 Failure of traditional step-size control

A key component of a robust covering algorithm is step-size control to enable adaptation to the local geometry of a branch. Typical methods are based on curvature,

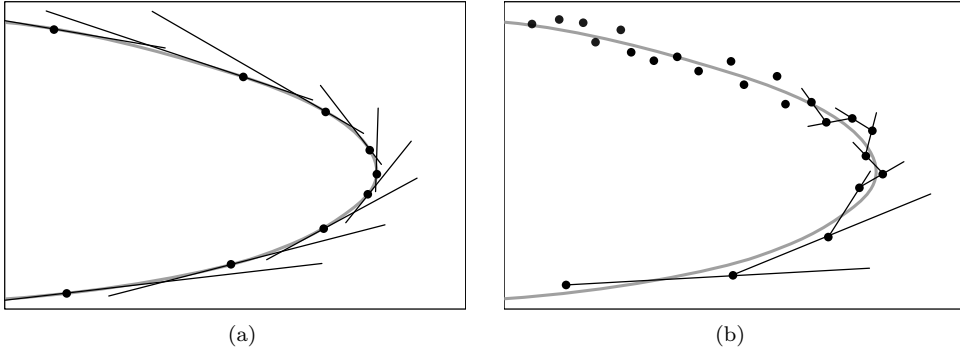


Figure 6: Adaptation of the continuation step-size based on bounding the angle between two successive continuation directions. In the case of smooth zero problems (a) one typically uses the tangent vector as continuation direction. The step-size is reduced if the local curvature increases, and increased again if the local curvature decreases. In the case of noise-contaminated zero problems (b) it is natural to use the vector through two successively computed points as a continuation direction. Here, we frequently observed that the step-size is *not* increased again after passing through a segment with higher local curvature, if the continuation step-size was reduced to a value comparable with the SD of the observation error.

for example, bounding the angle between two successive tangent vectors, or on counting the number of correction steps. Since for noise contaminated zero problems the number of correction steps can vary randomly, and we observed quite extreme variations in our experiments, the latter approach will lead to rather erratic changes of the step-size. Hence, our initial attempts were focused on applying curvature based step-size control.

We experimented with two different measures of curvature for step-size control: the angle between two successive continuation directions, and an estimate of the approximation error of a polynomial approximation, computed as the difference between two fits with polynomials of degree 2 and 3. Our experience can be summarized as follows: the method performs as expected as long as the step size is significantly larger than the SD of the observation error. As soon as the step-size has to be reduced to a value comparable with this SD due to high curvature, we observed frequently that the step-size control failed to increase the step-size again after leaving the region of high curvature; see Figure 6. The reason for this detrimental behavior is, that we found it impossible to implement a simple test that could reliably distinguish between a situation of high curvature and low SD, and a situation of high SD and low curvature. Another side effect of this phenomenon of stuck small step-size is, that the continuation path becomes highly jagged on the scale of the continuation step-size.

Since we found it impractical to apply an advanced algorithm for step-size control, we turned our attention to basic methods. As mentioned at the beginning of this section, while counting correction steps N and computing N/N_{tar} , where N_{tar} is the number of iterations targeted by step-size control, will result in rather random choices of step-sizes, the binary events of success and failure of the correction method are much less affected by noise. In addition, a failure of convergence must be handled in any case. The basic idea of using the events of success and failure for step-size control is that continuation tries to proceed with constant step-size, which is modified only in the event of failure of correction.

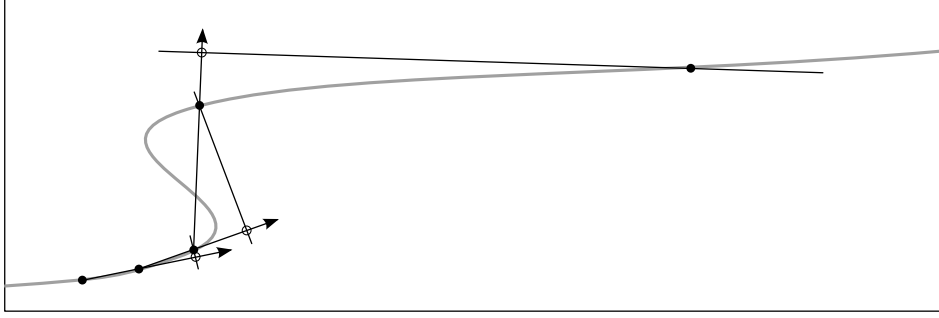


Figure 7: Sketch of the progress of continuation with constant step-size, secant direction as continuation direction and pseudo arc-length condition as projection condition. In particular, using a linear projection condition tends to show unwanted behavior when encountering certain geometric features with a size comparable to the (minimal) continuation step-size. For example, in our experiment we observed largely varying actual distances between consecutive points and suspect that this behavior is caused by small hysteresis loops along the branch.

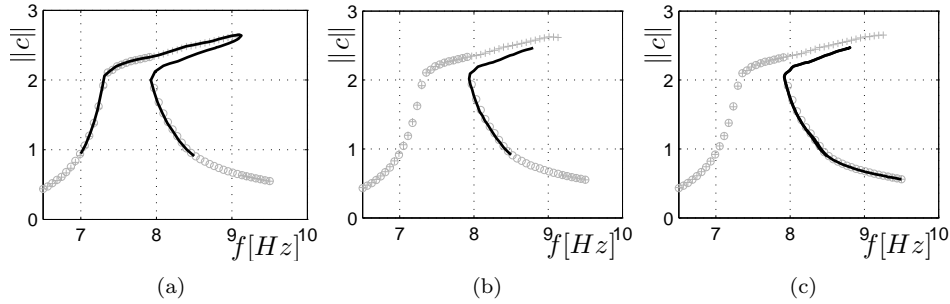


Figure 8: Sample of runs taken from a sequence of 9 runs. All runs started at the lower-right point at 8.5Hz. Only 4 out of all 9 runs succeeded (a). The reason for failure was that at certain frequencies the continuation often turned back and started recomputing the already covered part of the branch. The version of the code used for this sequence of runs employed a test for detecting a turn around. Therefore, the continuation would either terminate if a turn around was detected (b) or double cover in case the test failed to detect the turn around (c).

4.3 Occurrence of large jumps and double covering

Two observations we made frequently during experiments were the occurrence of continuation steps for which the distance between two successive points along a branch was several times larger than the continuation step-size as sketched in Figure 7, and the observation that the continuation would either start oscillating around a point without progressing, or turn back and cover the already computed part a second time; see Figure 8. In particular, the last problem occurred with such a frequency that it was observed in more than half of our continuation runs.

Although these observations seem unrelated, they are actually both caused by the choice of the continuation direction. Large jumps occur when a continuation direction that is almost perpendicular to the branch is chosen for some reason. This was observed in a region where we suspect the existence of a small hysteresis loop, see Figure 7, and can also occur in regions of higher SD of the observation error. Since we cannot exclude such choices of continuation directions, this problem was addressed by using non-linear projection conditions, which ensure that the actual

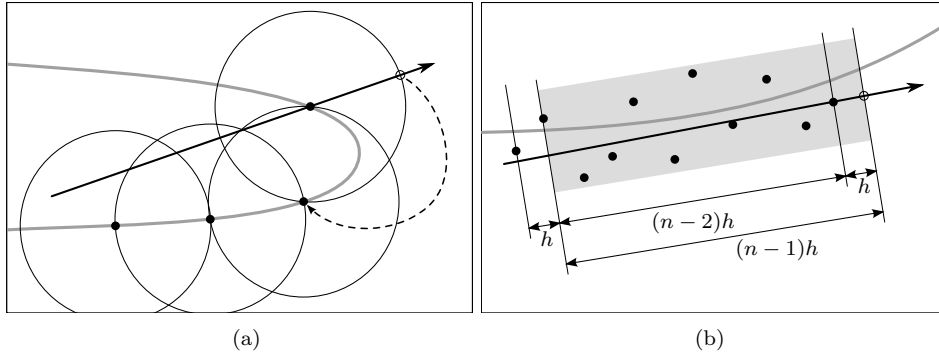


Figure 9: Shifting an approximation window containing n consecutively computed points (b) along with the continuation seems a way to allow using small continuation step-sizes while ensuring that the continuation direction changes slowly from step to step. On the other hand, this conservative choice of continuation direction can result in predicted points from which the correction converges to a point in backward direction (a). The sketch in (a) also illustrates that an averaged continuation direction counteracts the effect of a trust region around the predicted point. For the averaged direction shown, the point in backward direction is closer to the predicted point than the point in forward direction. Using the almost vertical secant direction instead would resolve this problem, a trust region would reject the solution point in backward direction.

distance between two consecutive points is bounded relative to the continuation step-size. A drawback of such conditions is, that a point in the backward direction is often also a solution, that is, the continuation might turn back depending on which point the corrector converges to; see Figure 9(a). We addressed the latter problem successfully by defining a spherical trust region around the predicted point, which prevents using points in backward direction; for technical details see Section 5.2.

The oscillatory behavior and the turning back of the continuation are mainly observed when using more than two points for estimating the continuation direction. The objective of using more points for estimating the continuation direction is to allow using small continuation step-sizes that are of the order of the SD of the observation error; see Figure 9(b). It may sound surprising that this fails, because it seems that if a continuation succeeds with step-size h using two points, it should also succeed with step-size $h/2$ using three points for estimating the continuation direction. However, this is not what we observed in experiments. Using two consecutive points only, our most recent algorithm typically succeeds, while using three or more points often leads to a situation where the corrector converges to a point in backward direction; see Figure 9(a). If higher accuracy and smaller step-sizes are required in an application, it is recommended to increase the number of points sampled inside the corrector to reduce the SD of the computed means, which will also reduce the influence of random fluctuations on the computed continuation directions and, therefore, allow for using smaller step-sizes.

4.4 Effects of slow drift of environmental parameters

A subtle effect we did not discover for a long time is caused by a slow drift of environmental parameters that are beyond the control of the experimentalist. While we observed visible differences between bifurcation diagrams recorded at different times of a day as illustrated in Figure 10, we did not expect that such differences would have any effect at the time scale of a prediction-correction step. By the time

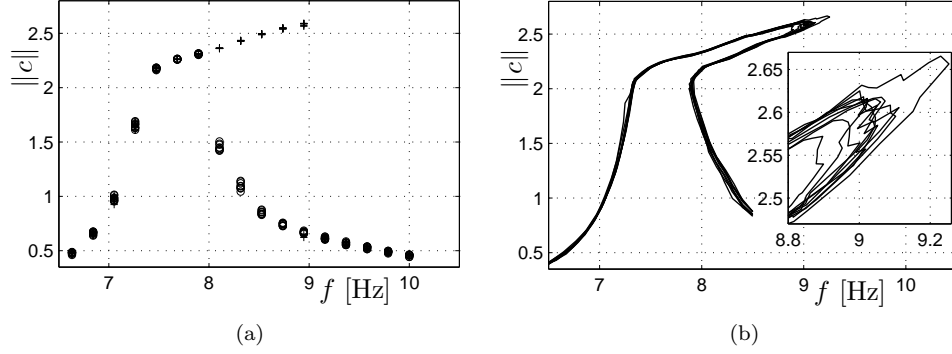


Figure 10: Drift of sweeps (a) and the bifurcation diagram (b) observed during a sequence of 9 consecutive runs, shown here as an overlay of plots. The start times of subsequent runs are spaced by approximately 100 minutes. While all branches match quite well for their largest parts, we observe a significant drift of the upper-right fold point of about 0.4Hz during the 15 hours of combined run time.

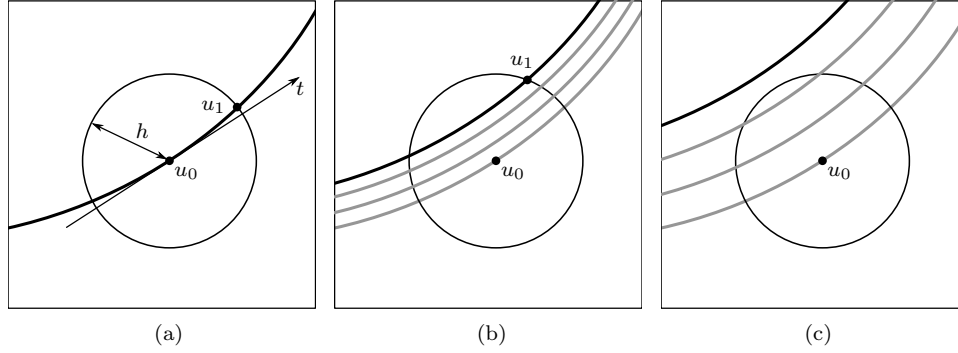


Figure 11: Due to a slow change of environmental parameters, a branch might drift away from its original location (a) during the time required for computing a new point. Continuation will succeed if the drift remains small (b), but might fail if the drift becomes too large (c).

of this discovery we were already using a non-linear projection condition to avoid large jumps as discussed in the previous section. Furthermore, we were testing a strategy in Newton's method for computing mean values of solution points with high statistical quality. Unfortunately, sometimes this correction process required several minutes of run time, which was apparently enough time for the solution branch to drift out of reach of the non-linear projection condition.

As an illustrative example of this effect let us use a non-linear projection condition that defines a sphere with radius h around the initial point u_0 ; see Figure 11(a). The point u_0 was produced by the covering algorithm in the previous continuation step and, hence, the sphere typically intersects with the branch in at least two points. The corrector will attempt to locate the point u_1 in forward direction. What we discovered was, that while the point u_0 computed at time T_0 was a solution at time T_0 , this did no longer hold true as the corrector progressed; see Figure 11(b). In fact, in certain situations the corrector required so much computation time that the branch drifted away from u_0 to a distance larger than h , which means that there was no solution any more and correction failed as a consequence; see Figure 11(c). Obviously, reducing the step size after correction failed is counterproductive in this

situation.

The impact of slow drift is particularly problematic if a correction step must be repeated. Typically, a correction step is considered unsuccessful if the number of iterations of the corrector exceeds a user-defined maximum, which might need to be quite large for noise-contaminated zero problems. Since in experiments the computation time of the corrector is dominated by the measurement time per correction step, reducing the number of unnecessarily executed correction steps will also reduce drift. Therefore, our key technique to address drift is an additional test that stops the corrector as soon as there is an indication that correction might not succeed, and to retry with different settings assuming that u_0 is still close enough to the branch we try to continue and that the modified settings increase the chances of success; see Sections 5.1 and 5.3 for details. A second technique to reduce the impact of slow drift is to use a non-linear pseudo arc-length condition that is tailored towards avoiding large jumps while still catching drifting branches. These two somewhat competing requirements lead to the invention of the egg-shape condition described in Section 5.2.

5 A continuation method for noise contaminated zero problems

The toolbox CONTINEX [12, 19] implements a predictor-corrector continuation method, which is tailored towards noise-contaminated zero problems. It proceeds as usual, starting from a known solution point and a vector defining the continuation direction, the predictor produces a new initial solution guess, which is subsequently corrected back to the branch by the corrector. An important modification to a traditional method is, that our method uses two modes of operation. The predictor initializes the mode of operation to *converge* and the corrector proceeds in *converge* mode until a first point is accepted. The corrector then changes the mode of operation to *sample* and continues to collect acceptable points until a statistically motivated condition is satisfied. The mean of the collected points is returned by the corrector as the new solution point along the branch. After successful computation of a new solution point the continuation direction is updated and the procedure is repeated. The detailed description of our method presented in this section follows the conventions introduced by COCO [13, 12], where a covering algorithm is composed of a zero problem together with an atlas, curve segment and corrector class.

5.1 An atlas class for noise-contaminated zero problems

The atlas class of the toolbox CONTINEX [12, 19] combines the functionality of the curve segment and corrector classes to a predictor-corrector pseudo arc-length covering algorithm with secant prediction. In addition, the atlas class implements step size control and handling of so-called *gaps*, which are regions on the solution branch where severe convergence problems are observed during correction.

Our algorithm for step size control follows the idea of using observations of the convergence of the corrector for adaptation of the step size. Typically, the number of iterations is used as an indication for adjustments, which works well for problems with sufficient smoothness. For noise contaminated zero problems, this translates to monitoring an estimate of the expected value of iterations, which is unnecessarily expensive. Therefore, we take only the events of success and failure of the corrector into account, which requires only a single execution of the corrector in each continuation step. Our algorithm is simple, we attempt to execute every continuation step with the maximum step size h_{\max} and modify the step size only

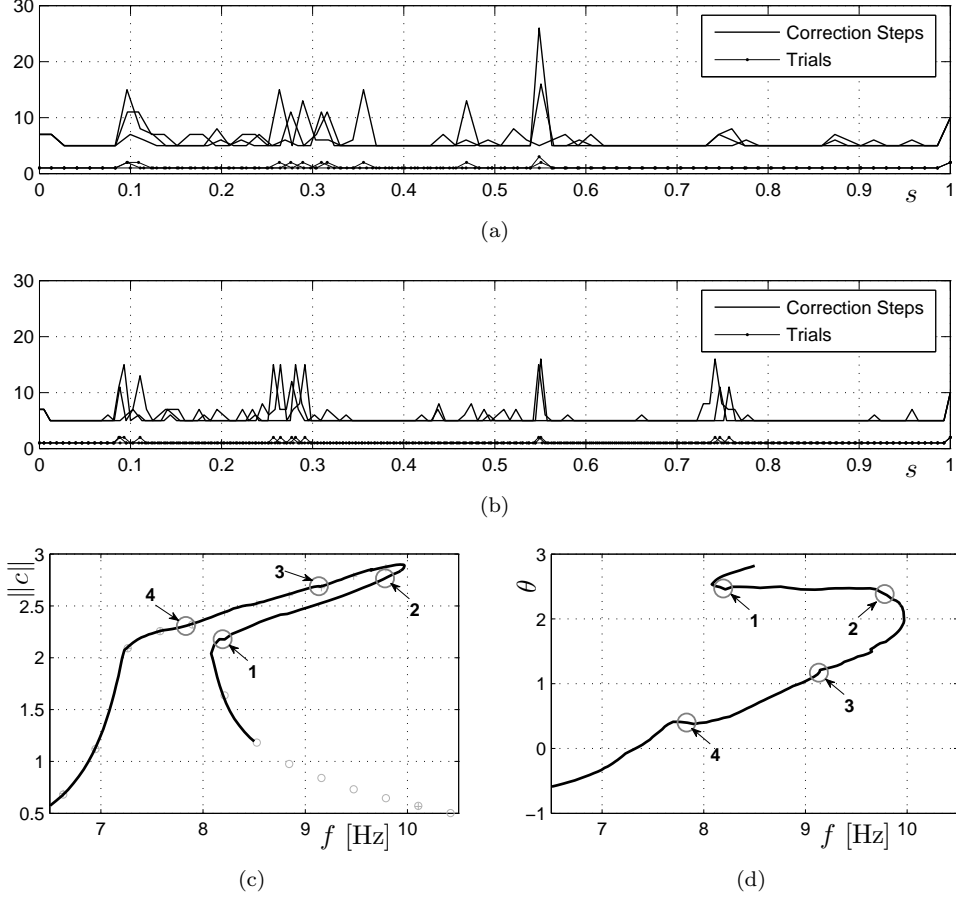


Figure 12: Illustration of the performance of failure-based step-size control. Panels (a) and (b) show the number of trials and the total number of correction steps for continuation runs with $h_{\max} = 0.2$ and $h_{\max} = 0.1$, respectively. In both cases, continuation proceeded along the branch shown in (c) and (d) in two different projections, as a frequency-response plot in (c) and a frequency-phase plot in (d). The continuation started at the endpoint at 8.5Hz. The data in each panel is collected from 3 different runs with coarser (a) and finer (b) step size. To allow a comparison, the data is plotted as a function of normalized arc-length. We observe consistently that in all runs a larger number of correction steps was executed near $s = 0.1$ (label 1), $s = 0.275$ (label 2), $s = 0.55$ (label 3) and $s = 0.75$ (label 4). These points are highlighted in panels (c) and (d) to illustrate their location in the bifurcation diagram. The phase θ in (d) is computed as $\text{atan2}(c_2, c_7)$, where c_2 and c_7 are the first-order sin and cos components of the Fourier modes (1), respectively.

in the case that correction fails. This will result in a covering of the solution branch with points mostly equi-distributed with respect to pseudo arc-length and some regions with a denser or coarser mesh. Figure 12 illustrates this strategy with data collected in 6 different runs, of which 3 runs were executed with a coarser step size of $h_{\max} = 0.2$ and the other 3 runs with the finer step size $h_{\max} = 0.1$. For most points the corrector converges quickly with minimal number of iterations, while in some areas a larger number of iterations or even a second trial are necessary.

Our atlas algorithm will attempt up to N_{trial} executions of a predictor-corrector step, where for each trial a user can specify a different set of parameters for the corrector algorithm, for example, different tolerances. In addition, one can specify a different prediction step-size for each trial, as well as whether or not an accepted point should be marked as a gap, that is, as a point with insufficient accuracy that is accepted only for the purpose of being able to proceed with the continuation in some way. Reduction and increase of the continuation step-size are implemented via re-scaling of the reference step-size used by the curve segment class. Our covering algorithm initially uses a curve segment with reference step size $h = h_{\max}$ and, unless specified otherwise, uses the default scaling factors $s_k = 1 - \lambda_k(1 - h_{\min}/h_{\max})$, $\lambda_k = (k - 1)/(N_{\text{trial}} - 1)$, for each trial $k = 1, \dots, N_{\text{trial}}$ of correction. That is, the default scaling sequence results in a linear down-scaling from h_{\max} to h_{\min} for each trial.

A peculiarity of solution branches defined by noise affected zero problems is that correction with higher accuracy might fail at small subsets of the solution branch for various reasons. Since, in particular, in experiments it is very expensive to terminate a continuation run because of temporary convergence problems, we try to pass through regions of poor convergence by using a scaling factor of $s_{N_{\text{trial}}} = 2$ for the last trial together with very loose conditions for accepting points by the corrector. The expectation is that continuation will resume with full accuracy after just a few such steps. Our covering algorithm allows to mark points in such regions as gap points, and keeps track of how many consecutive points were marked as a gap point. For example, points accepted in our final attempt with $s_{N_{\text{trial}}} = 2$ are marked as gap points. If the number of consecutive gap points exceeds a predefined maximum the continuation ultimately terminates.

As a consequence of this strategy of proceeding by all means, one ought to investigate the convergence behavior along a branch after data collection. For example, points marked along the branch with labels 1-4 in Figure 12 correspond to locations where correction consistently required a larger number of iterations or even an additional trial. Investigating different representations of a solution branch might provide clues as to why one observed slower convergence near some points. For example, near the points 1 and 3 we observe small kinks that indicate that slow convergence might be caused by the local geometry of the branch. On the other hand, at points 2 and 4 the branch seems to be rather smooth, indicating that in these regions slow convergence might be due to peculiar effects not reflected in the bifurcation diagram. Here, further experiments under human supervision seem warranted, while for the remaining parts of the bifurcation diagram the behavior of the impact oscillator experiment seems to be captured accurately.

5.2 A curve segment class with non-linear arc-length condition

The curve segment class of the toolbox CONTINEX [12, 19] implements a secant approximation of a curve segment to the solution branch and secant prediction, which are simple algorithms for interpolation and prediction requiring only weak assumptions on existence and regularity of solutions. Essentially, if the error in the estimation of the mean value of a solution point provided by the corrector is

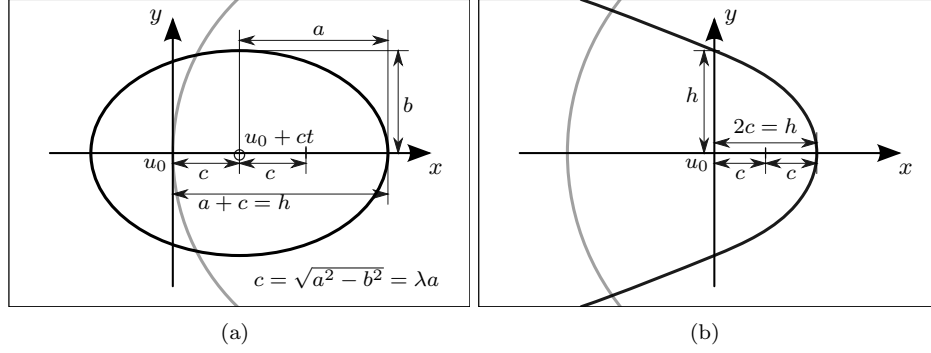


Figure 13: Generating curves of the ellipse (a) and parabola (b) shapes. The continuation direction is aligned with the x -axis and the gray arc indicates the boundary of the trust region. Only points to the right of this boundary will be accepted. The eccentricity of the generating ellipse can be modified by choosing the parameter λ .

small compared with the continuation step size, this method will succeed if the zero set of the projection condition intersects the noise-contaminated solution branch transversally. Furthermore, the curve segment class provides three different non-linear projection conditions, which define rotationally symmetric hyper surfaces we refer to as *shapes*. The symmetry axis of each shape is the line in continuation direction through the starting point.

Denote with $x := t^T(u - u_0)$ the orthogonal projection of u onto t and based at u_0 , with $y := u - u_0 - xt$ the projection onto the hyper plane through u_0 and normal to t , and with h the continuation step size. Our non-linear projection conditions can then be written in the general form $G(x, y^T y) = 0$, which implies rotational symmetry with respect to the x -axis. In the case that $u \in \mathbb{R}^2$ this equation defines a planar curve that we refer to as the *generating curve* and we use the name of a generating curve to refer to its corresponding shape. Furthermore, we will below use a representation of generating curves in Cartesian coordinates (x, y) that is straightforward to translate into a representation in the coordinates above using the substitution $(x, y^2) \mapsto (x, y^T y)$. As an important technical detail, the specific definition G of a shape was selected to ensure that the derivative of $s \mapsto G(sx, (sy)^T(sy))$, $s \in \mathbb{R}$, is of order 1 for any choice of x and y , if possible. That is, the surface defined by $G(x, y^2) = z$ in the coordinates $(x, y, z) \in \mathbb{R}^3$ is approximately conical with slope 1.

For each shape the curve segment class defines a *pull-back map* and a radius for a *trust region*. A trust region is a ball around the predicted point and used to restrict the iteration process of the corrector to a neighborhood of the predicted point. The primary aim of the trust region is to restrict function evaluations to a region of safe operation of an experiment and to prevent accidental branch jumping in case the controlled experiment is not globally unique as outlined in Section 3. A secondary goal is to avoid that points in direction $-t$ from u_0 are accepted. A pull-back map computes for any given point a unique point on a shape. In other words, it “pulls” a point not on the shape “back” onto the shape. Pull-back maps are used by the corrector and their application effectively eliminates the additional non-linearity introduced by the shape functions by explicitly restricting the iterates of the corrector to the shape.

Ellipse-shapes are generated by an ellipse with numerical eccentricity $c/a = \lambda$ and major half axis $a := h/(1 + \lambda)$, where $0 \leq \lambda < 1$ is a parameter chosen by the user; see Figure 13(a). For $\lambda = 0$ we obtain a circle with radius h and for $\lambda > 0$ an

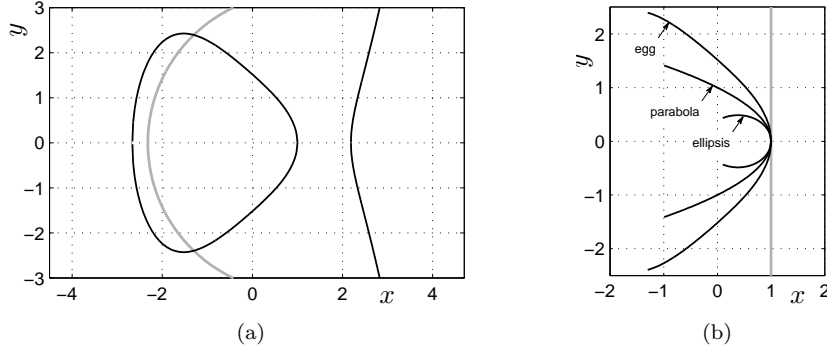


Figure 14: The egg-shape without the case distinction (6) has two disconnected components (a), the case distinction removes the component located in the half-plane $x > 2$ while preserving differentiability. The continuation direction is aligned with the x -axis and the gray arc indicates the boundary of the trust region. Only points to the right of this boundary will be accepted. Panel (b) shows a comparison of all non-linear shapes together with the ‘line-shape’ corresponding to the pseudo arc-length condition. The shapes in (b) are restricted to their respective trust regions. In this coordinate system, the predicted point is located at $(1, 0)$ and the base point u_0 of the continuation step at the origin $(0, 0)$.

ellipse with u_0 located in its left-hand focal point. In our implementation we use the inverse golden mean $\lambda = 2/(1 + \sqrt{5}) \approx 0.618$ as default, which results in a good balance between eccentricity and maximum curvature and was observed to perform well as a non-linear projection condition.

The ellipse-shape is defined as

$$G(x, y^2) := \sqrt{\frac{b}{a}(x - c)^2 + \frac{a}{b}y^2} - \sqrt{ab}. \quad (3)$$

As trust region we use the ball $(x - h)^2 + y^2 \leq h^2$ with radius h equal to the continuation step size. The pull-back map of the ellipse-shape is defined as the intersection of the vector $d := u - (u_0 + ct)$ with the shape, given by $u \mapsto (u_0 + ct) + sd$, where

$$s := \sqrt{\frac{ab}{\frac{b}{a} \cos^2 \alpha + \frac{a}{b} \sin^2 \alpha}}. \quad (4)$$

In our implementation we exploit the equalities $\cos \alpha = t^T(u - u_0)/\|u - u_0\|_2$ and $\sin^2 \alpha = 1 - \cos^2 \alpha$.

The parabola-shape is generated by a parabola chosen to satisfy the two conditions $y(h) = 0$ and $y(0) = \pm h$. The shape is defined as

$$G(x, y^2) := h(x - h) + y^2 = 0, \quad (5)$$

and we chose the trust region $(x - h)^2 + y^2 \leq 6h^2$ such that its boundary intersects the parabola at $(x, y^2) = (-h, 2h^2)$. The pull-back map of the parabola-shape is defined as the intersection of the vector $d := u - u_0$ with the shape, given by $u \mapsto u_0 + (1/s)d$, where s is the positive solution of the quadratic equation $h(x/s - h) + (y/s)^2 = 0$. Note that we use reciprocal scaling for a numerically stable computation of s .

The egg-shape is generated by the curve $\sqrt{x^2 + y^2} = r(x/h)$, where

$$r(z) := 1 + \begin{cases} 0 & z \geq 1, \\ c_0 + c_1 z + c_2 z^2 + c_3 z^3 & z < 1, \end{cases} \quad (6)$$

$c = (1/27)(14, -24, 6, 4)$, is a distance function with the properties $r(1) = 1$, $r'(1) = 0$, $r(-2) = 2$, $r'(-2) = 0$. These properties ensure that the interior of the egg-shape is a convex set and that the radius of curvature at $z = 1$ is equal to 1. Furthermore, r is continuously differentiable. As trust region we use the ball $(x - h)^2 + y^2 \leq 3.32h$, where the factor 3.32 was chosen after inspection of the graphical representation; see Figure 14(a). The pull-back map of the egg-shape is defined as the intersection of the vector $d := u - u_0$ with the egg-shape, given by $u \mapsto u_0 + sd$, where the scaling factor s is computed iteratively with a subdivision algorithm.

Note the trade-off between a simple explicit expression for the derivative of the shape function and a simple pull-back formula. We use a simple expression for the radius function r in Cartesian coordinates, which gives a simple expression for the derivative, but requires an approximation method for computing the pull-back map. However, since experiment and continuation run asynchronously and the run time is determined by measurements and not computations, this trade-off seems of little practical relevance. If necessary, one could replace the iteration method with a look-up table defining the pull-back radius as a function of $\cos \alpha$ as for the ellipse-shape.

The ellipse-shape results in a behavior that a user would expect intuitively. If the local curvature of the solution branch is high, a smaller continuation step size is used, because the length of the line l from u_0 to a point on the ellipse is a strictly monotonically decreasing function of the angle between l and t . Conversely, for the egg-shape this length is strictly monotonically increasing, which might be counter intuitive. However, this holds true for the pseudo arc-length condition as well. In fact, the egg-shape leads to considerably smaller distances between consecutive points than the pseudo arc-length condition. The parabola-shape is a compromise between the ellipse- and the egg-shape. However, our experience indicates that this shape might be useful in exceptional situations only. For smooth problems with no or negligible noise the ellipse-shape performs best. For problems with considerable noise levels, and/or considerable drift of solutions over time, the egg-shape is the shape of choice; see Figure 14(b) for a graphical comparison.

A curve segment selects a projection condition depending on the mode of operation of the covering algorithm. While in converge mode one of the non-linear projection conditions defined above is used, in sample mode we always use the affine linear projection condition

$$g^T(u - \tilde{u}) = 0, \quad (7)$$

where \tilde{u} is a point provided by the corrector, and g is the normalized gradient vector of the shape surface function at \tilde{u} . That is, g is a vector normal to the shape at the point \tilde{u} and any u satisfying (7) lies in a hyper plane tangent to the shape. Furthermore, in sample mode the identity is used as pull-back map. Using (7) as projection condition and the identity as pull-back map ensures that all subsequently accepted points are samples of a distribution with mean value on the hyperplane defined by (7).

5.3 A modified damped Newton method

The corrector class of the toolbox CONTINEX [12, 19] implements a variation of the damped Newton method $u_k = u_{k-1} - \gamma_k d_k$, $k = 1, 2, \dots$, where $0 < \gamma_k \leq 1$ is a damping factor and the Newton correction d_k is computed as the solution of a system of linear equations $J_{k-1} d_k = F(u_{k-1})$. This iteration is usually repeated until either $\|d_k\|_2$ is smaller than a prescribed tolerance, or the number of iterations exceeds a certain value. The damping factor in step k is computed as $\gamma_k = \gamma \alpha^m$, where γ and α are some positive real numbers and the exponent m is either the smallest

integer $0 \leq m \leq m_{\max}$ such that the descent condition $\|F(u_k)\|_2 \leq \|F(u_{k-1})\|_2$ is satisfied, or $m = m_{\max}$, for some choice of m_{\max} . If J_{k-1} is the linearization of F at u_{k-1} we obtain Newton's method and Newton-like methods otherwise. In our experiments we typically use $\gamma = 1$, $\alpha = 1/2$ and $m_{\max} = 4$. As outlined in Section 5.4, our zero problem provides a matrix J_{k-1} that is the result of Broyden updates of a finite-difference approximation.

Our main modifications to this basic algorithm are (A) the introduction of *a-priori* damping, (B) the restriction of iterates to the intersection of a trust region with a trusted set, (C) using a pull-back operation $\text{pbm} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ that ensures that some equations of the zero problem are satisfied exactly, (D) collecting a sample Σ of acceptable points, and (E) statistically motivated stopping conditions. A sketch of the resulting modified Newton method is

1. Set $\Sigma = \emptyset$, $k = 0$, **mode** = converge.
2. Set $k = k + 1$, $\gamma = 1$; evaluate $F_{k-1} = F(u_{k-1})$; compute $J_{k-1} \approx F'(u_{k-1})$; solve $J_{k-1}d_k = F_{k-1}$ for d_k .
3. Reduce γ such that $\|\gamma d_k\|_2$ is sufficiently small compared with $\|u_{k-1}\|_2$.
4. Further reduce γ such that $\text{pbm}(u_{k-1} - \gamma d_k)$ is an element of a trusted set.
5. For $m = 1, \dots, m_{\max}$ do
 - (a) Compute $u_k = \text{pbm}(u_{k-1} - \gamma d_k)$.
 - (b) If u_k inside trust region, go to step 7.
 - (c) Further reduce γ : set $\gamma = \alpha\gamma$.
6. Return with error.
7. For $m = m, \dots, m_{\max}$ do
 - (a) Compute $u_k = \text{pbm}(u_{k-1} - \gamma d_k)$.
 - (b) If $\|d_k\|_2 \leq \text{TOL}$ and $\|\text{mean}(\{F(u_k)\} \cup F(\Sigma))\|_2 \leq \text{ResTOL}$, go to step 9.
 - (c) If $\|\text{mean}(\{F(u_k)\} \cup F(\Sigma))\|_2 \leq \|\text{mean}(\{F(u_{k-1})\} \cup F(\Sigma))\|_2$, go to step 8.
 - (d) Further reduce γ : set $\gamma = \alpha\gamma$.
8. Check stopping conditions and either repeat at step 2 or return with error.
9. Set **mode** = sample; add u_k to Σ ; check stopping conditions and either return $u^* = \text{mean}(\Sigma)$ as solution or repeat at step 2.

The *a-priori* damping (A) is implemented in step 3, the restriction (B) in steps 4 and 5, the sampling (D) is used in steps 7b, 7c and 9, the stopping conditions (E) in steps 8 and 9, and the pull-back map (C) whenever a new iterate is computed. Note that this Newton algorithm does not stop after the first point has been accepted, but rather continues until a sufficiently large number of acceptable points has been collected in Σ . The mean of these collected points is then returned as an approximation to the mean of the solution.

The quite elaborate sequence of preparation steps 3 to 5 aims at producing starting values for m and λ , $m \geq 1$ as small as possible and $\lambda \leq 1$ as large as possible, such that the loop in step 7 is still executed only once. To give a precise description of these steps, define the scaling function $z = \text{scl}(x, y)$, $\text{scl} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, as the function that returns a vector z with components $z_i := y_i / (1 + |x_i|)$, that is, z is a

component-wise scaling of y relative to the elements of x . With this function, the *a-priori* damping in step 3 can be written as $\gamma \mapsto \min\{\gamma, S_{\max}/r\}$, where

$$r := \frac{\|\text{scl}(u_{k-1}, d_k)\|_2}{1 + \|\text{scl}(u_{k-1}, u_{k-1})\|_2} \quad (8)$$

is a relative Newton step size, and the maximum relative Newton step size $S_{\max} > 0$ is a real number chosen by the user. In other words, the *a-priori* damping is either the identity if $\gamma r \leq S_{\max}$, that is, for sufficiently small $\|d_k\|_2$ compared with $\|u_{k-1}\|_2$, or a suitable downscaling of the damping factor γ . We usually use $S_{\max} = 1/10$.

In step 4 we further restrict γ such that the point $\text{pbm}(u_{k-1} - \gamma d_k)$ is an element of a trusted set. The rationale behind this restriction is to avoid correction steps that are too large for an experimental setup to handle. We also observed that restricting the size of correction steps aggressively lead to a more uniform convergence and more robustness to noise in our experiments. Let U be a finite non-empty set of points in \mathbb{R}^n . The trusted set

$$V(U, \delta) := \{x \in \mathbb{R}^n \mid \min_{y \in U} (\|x - y\|_2) \leq \delta\} \quad (9)$$

is the union of closed balls with radius $\delta > 0$ around each point in U . To obtain a suitable value for γ we apply the map

$$\gamma \mapsto \sigma \gamma \delta / \min_{y \in U} (\|\text{pbm}(u_{k-1} - \gamma d_k) - y\|_2) \quad (10)$$

repeatedly until $\text{pbm}(u_{k-1} - \gamma d_k) \in V(U, \delta)$. Here, $0 < \sigma < 1$ is a safety factor ensuring termination of the repetition after a finite number of steps if $u_{k-1} \in U$. In our implementation we always include u_{k-1} and all accepted points Σ in the trusted set and a user can choose to include a larger number of past iterates. We usually set $\sigma = 0.95$ and the distance δ is a fraction D/N of the radius D of the trust region, where N is typically an element of $\{4, 5, 6\}$.

In step 5, before executing the common damping method, we ensure that the point $\text{pbm}(u_{k-1} - \gamma d_k)$ is an element of a trust region. In our case, the trust region is simply the ball $B(u_0, D)$ with radius D around the initial point u_0 and represents the expectation that the solution cannot lie further away from u_0 than D . In other words, if the Newton iterations leave the trust region, we assume that this attempt of approximating a solution failed. To obtain a suitable value for γ , we apply the map $\gamma \mapsto \alpha \gamma$ repeatedly until either $\text{pbm}(u_{k-1} - \gamma d_k) \in B(u_0, D)$ or the number m of repetitions exceeds m_{\max} . If the resulting point $\text{pbm}(u_{k-1} - \gamma d_k)$ is not an element of the trust region, our Newton method stops with an error.

After this preparation we execute in step 7 the traditional damping steps as explained at the beginning of this section, continuing with the count m and the value of γ resulting from step 5. The only modifications are the application of the pull-back map in 7a, the stopping condition in 7b, and the descend condition in 7c. In each iteration we check whether or not the two conditions $\|d_k\|_2 \leq \text{TOL}$ and $\|\text{mean}(\{F(u_k)\} \cup F(\Sigma))\|_2 \leq \text{ResTOL}$ for accepting a point are satisfied simultaneously, where TOL and ResTOL are user-defined tolerances and $\text{mean}(\{F(u_k)\} \cup F(\Sigma))$ is computed as

$$\text{mean}(\{F(u_k)\} \cup F(\Sigma)) := \frac{1}{N+1} \left(F(u_k) + \sum_{i=1}^N F(u_{k_i}) \right), \quad (11)$$

where N is the number of previously accepted points and k_i , $i = 1, \dots, N$, are the indices of the accepted points collected in Σ . Consequently, our corrector algorithm

accepts and collects any point for which the mean of the residues remains below a prescribed tolerance, that is, if some accepted points have very small residuum, points with a residuum somewhat larger than ResTOL become acceptable as well. The condition on the mean value is motivated by the property of linear functions that the residuum of the mean is equal to the mean of the residues, and well-justified in the case that the mean value of F can locally be approximated by a linear function.

The requirement of satisfying two conditions simultaneously might sound restrictive. However, in our experiments we found it very useful to reject points resulting from a small Newton step but having large residuum, and points having small residuum but resulting from a larger and possibly random jump. In other words, the combined condition helps eliminating false positives. For problems with no or negligible noise one of the conditions is typically redundant. Setting one of the tolerances to `Inf` will disable the test.

Differently to a traditional Newton method, our method does not terminate in step 9 upon acceptance of a single point, but rather changes the working mode to sample mode, inserts the point into the set Σ of accepted points, and continues iterating in sample mode until sufficiently many acceptable points have been collected. The first accepted point u_{k_1} is used as the base point \tilde{u} in the projection condition (7). As a reminder, operating in sample mode means that for all remaining iterations the curve segment class applies an affine-linear projection condition for which the mean of any finite subset of solutions is a solution as well.

Whenever a point is accepted in step 9, this point is inserted into the collection Σ of accepted points and we test if the stopping condition

$$\left| \{i \in \mathbb{N} \mid \|u_{k_i} - u^*\|_2 \leq \varrho, i = 1, \dots, N\} \right| \geq N_{\min}, \quad (12)$$

where $|A|$ denotes the cardinality of the set A , is satisfied. That is, we stop the iteration process if the number of accepted points that lie within a ball of radius ϱ around the mean value $u^* = \text{mean}(\Sigma) = (1/N)(u_{k_1} + \dots + u_{k_N})$ is larger or equal to the value N_{\min} . The mean value u^* is then returned as the result of the correction step.

In steps 8 and 9 our algorithm uses in total three conditions for termination, the first of which is returning in step 9 after successful computation of a collection Σ satisfying (12). Secondly, our method terminates with an error in step 8, if

$$k + \max(N_{\min} - N, 0) > k_{\max}, \quad (13)$$

where $N = |\Sigma|$ is the current collection size, N_{\min} is the minimum collection size used in (12) and k_{\max} is the maximum number of correction steps allowed. That is, we terminate the computation as soon as it becomes impossible to collect at least N_{\min} points.

The final condition for termination, also used in step 8, is

$$k - k_{\min} > N \text{ceil}((k_{\max} - k_{\min})/N_{\min}), \quad (14)$$

where k_{\min} and k_{\max} are the minimum and maximum number of correction steps to be executed and the function $\text{ceil}(x)$ computes the smallest integer larger than or equal to x . This condition requires for successful correction that, after an initial phase of k_{\min} correction steps, points are collected in Σ with a certain minimum frequency. This requirement aims at stopping the correction as soon as a low rate of collecting points indicates that correction might not succeed with the current settings in order to enable a subsequent attempt of correction with different settings as fast as possible. Forcing a fast termination in such a case is motivated by avoiding a large drift of the branch during a prediction-correction step.

5.4 A zero problem for continuation in experiments

The toolbox CONTINEX [12, 19] provides an abstract framework for communication with experiments and simulations, and a number of more specialized classes for communication with SIMULINK models (synchronous and asynchronous), experiments connected to a dSPACE real-time control board (asynchronous), and simulations using MATLAB's ODE suite (synchronous). The abstract framework is suitable for continuation in experiments as well as continuation of solutions of so-called equation-free models. Here, we will focus on the abstract framework and refer to the documentation and examples shipped with CONTINEX [12, 19] for details about the classes for specialized types of problems.

Constructing an instance of a continex zero problem requires a handle to a function $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ returning a residuum vector given an input vector $x \in \mathbb{R}^n$ and a vector of parameters $p \in \mathbb{R}^m$, and an initial guess of a solution point $u_0 = (x_0, p_0)$. The evaluation of the function F may not be unique, that is, repeated evaluations of F with the same argument vectors may result in different residuum vectors due to uncertainties or an evaluation process affected by noise. Upon successful construction, an instance of a continex zero problem will provide for the evaluation of F and its linearization, an optional scaling of variables and residues, and caching of function values to avoid redundant evaluation for efficiency.

We will not comment on the evaluation of F , which is simply delegated to the function provided by the user. Furthermore, we will only give some general guidelines for scaling, which is otherwise largely problem dependent. Since for noise- or uncertainty-affected function evaluations it is usually not possible to require high accuracies in the corrector, it is often necessary to scale individual components of the input and output vectors at the interface between continuation and function evaluation. The scaling should be such that the most relevant components of the solution vector are of a reasonable order of magnitude compared with the tolerances prescribed in the corrector. If possible, the scaling also ensures that the noise level is of comparable order for all components. The functionality provided by the continex zero problem allows for scaling of vectors in a transparent way, that is, without the need to modify an experiment or model.

The linearization of the continex zero problem is computed in a two-step algorithm: the Jacobian of F is initialized using a one-sided finite-difference method, which allows to use different finite-difference step sizes for the variables x and parameters p to account for different sensitivities with respect to these variables. After initialization, the Jacobian is updated using Broyden updates until the number of computations of the Jacobian since initialization exceeds a user-defined limit. If this limit is exceeded, the Jacobian is re-initialized with a finite-difference approximation and the whole procedure repeats until the user-limit is exceeded again. As an important exception, the re-initialization is suspended in sample mode. This ensures that the sampling process in Newton's method is not affected by drift, which could occur due to the significant computational expense of producing a finite-difference approximation of the Jacobian compared with Broyden updates, which require no additional computational effort, because we can re-use data that was already computed at the beginning of a Newton step.

6 Conclusions: the toolbox CONTINEX and future directions

All algorithms described in this text are implemented in the toolbox CONTINEX [12, 19], which is part of the project COCOTOOLS [12]. The toolbox includes an interface class as a general definition of instances of noise contaminated zero problems, as well

as specialized implementations demonstrating the use of DSPACE, SIMULINK and ODE models. To define a new problem, a user can make use of these predefined interfaces or derive a subclass from our generic interface class. As a production example, all MATLAB files we developed for our experiment and all data obtained in the course of this project can be downloaded from the project site [19].

The current version of CONTINEX [12, 19] implements a robust covering method for regular solution branches of noise contaminated zero problems. Immediate directions for further research are the treatment of special points, for example, bifurcation points and singular points. Traditional continuation methods usually locate special points in a post-processing step after locating two regular points enclosing the special point. Our observations suggest that for noise contaminated zero problems it is necessary to anticipate the approach of certain special points, because it seems not possible to step over such points in a straightforward way. Anticipating the location of a special point might allow a suitable local modification of the covering algorithm such that it becomes possible to pass through the special point with high certainty. As subsequent steps, methods for switching branches at special points, and for computing branches of special points will be of interest.

Our interface to the special case of control based continuation implements several explicit stability tests; see [2]. These tests require significant additional measurement time and an interesting question is the development of methods that do not require additional measurements. For example, applying techniques commonly used for system identification [20] might provide more efficient means of assessing stability; see also [21].

Furthermore, during the execution of our Newton-like corrector algorithm we observe random behavior of the condition number of the sequence of matrices generated with Broyden updates. Sometimes, the condition number assumes an exceptionally large value for an iteration step, resulting in a large perturbation of convergence of the correction. Looking at the formula for Broyden updates,

$$J_i = J_{i-1} + \frac{(F_i - F_{i-1}) - J_{i-1}(x_i - x_{i-1})}{(x_i - x_{i-1})^T (x_i - x_{i-1})} (x_i - x_{i-1})^T \quad (15)$$

it is clear that the updates become very susceptible to noise as soon as the vectors F_i , F_{i-1} and $x_i - x_{i-1}$ are of the order of the noise level, that is, essentially random values carrying comparably little information about the zero problem. To obtain a more uniform behavior of the correction method, it seems promising to study the application of methods from time series analysis to construct a robust Broyden method that can compensate for random fluctuations in the sequence of function evaluations used for updates, for example, by applying suitable filters [20].

7 Acknowledgements

This work was supported by the Danish research council (FTP) under the project number 0602-00753B.

References

- [1] E. Bureau, F. Schilder, I. Santos, J. Thomsen, J. Starke, Experimental bifurcation analysis of an impact oscillator - tuning a non-invasive control scheme, *Journal of Sound and Vibration* 332 (22) (2013) 5883 – 5897. doi:10.1016/j.jsv.2013.05.033.
- [2] E. Bureau, F. Schilder, M. Elmegård, I. F. Santos, J. J. Thomsen, J. Starke, Experimental bifurcation analysis of an impact oscillator - determining

- stability, *Journal of Sound and Vibration* 333 (21) (2014) 5464 – 5474.
doi:<http://dx.doi.org/10.1016/j.jsv.2014.05.032>.
URL <http://www.sciencedirect.com/science/article/pii/S0022460X14004416>
- [3] J. Sieber, B. Krauskopf, Control based bifurcation analysis for experiments, *Nonlinear Dynamics* 51 (3) (2008) 365–377. doi:10.1007/s11071-007-9217-2.
URL <http://dx.doi.org/10.1007/s11071-007-9217-2>
- [4] J. Sieber, A. Gonzalez-Buelga, S. A. Neild, D. J. Wagg, B. Krauskopf, Experimental continuation of periodic orbits through a fold, *Phys. Rev. Lett.* 100 (2008) 244101. doi:10.1103/PhysRevLett.100.244101.
URL <http://link.aps.org/doi/10.1103/PhysRevLett.100.244101>
- [5] D. A. W. Barton, S. G. Burrow, Numerical continuation in a physical experiment: Investigation of a nonlinear energy harvester, *Journal of Computational and Nonlinear Dynamics* 6 (2010) 011010–011010–6. doi:10.1115/1.4002380.
URL <http://dx.doi.org/10.1115/1.4002380>
- [6] J. Sieber, B. Krauskopf, D. J. Wagg, A. Gonzalez-Buelga, S. A. Neild, Control-based continuation of unstable periodic orbits, *Journal of Computational and Nonlinear Dynamics* 6 (2010) 011005–011005–9. doi:10.1115/1.4002101.
URL <http://dx.doi.org/10.1115/1.4002101>
- [7] I. G. Kevrekidis, G. Samaey, Equation-free multiscale computation: Algorithms and applications, *Annual Review of Physical Chemistry* 60 (1) (2009) 321–344.
arXiv:<http://www.annualreviews.org/doi/pdf/10.1146/annurev.physchem.59.032607.093610>.
- [8] Y. Kevrekidis, G. Samaey, Equation-free modeling, *Scholarpedia* 5 (9) (2010) 4847.
- [9] O. Corradi, P. Hjorth, J. Starke, Equation-free detection and continuation of a Hopf bifurcation point in a particle model of pedestrian flow, *SIAM Journal on Applied Dynamical Systems* 11 (3) (2012) 1007–1032.
- [10] C. Marschler, J. Sieber, R. Berkemer, A. Kawamoto, J. Starke, Implicit methods for equation-free analysis: Convergence results and analysis of emergent waves in microscopic traffic models, *SIAM Journal on Applied Dynamical Systems* 13 (3) (2014) 1202–1238. arXiv:<http://dx.doi.org/10.1137/130913961>, doi:10.1137/130913961.
URL <http://dx.doi.org/10.1137/130913961>
- [11] Y. Kevrekidis, G. Samaey, Equation-free modeling, *Scholarpedia* 5 (9) (2010) 4847, revision #91237.
- [12] F. Schilder, H. Dankowicz, Continuation core and toolboxes (coco), SourceForge.net, project cocotools.
- [13] H. Dankowicz, F. Schilder, Recipes for continuation, Vol. 11 of Computational Science & Engineering, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013. doi:10.1137/1.9781611972573.
URL <http://dx.doi.org.globalproxy.cvt.dk/10.1137/1.9781611972573>
- [14] Dynamical systems software list on DS Web., <http://www.dynamicalsystems.org/sw/sw/>.

- [15] E. Doedel, H. B. Keller, J. P. Kernevez, Numerical analysis and control of bifurcation problems (i): Bifurcation in finite dimensions, *International Journal of Bifurcation and Chaos* 01 (03) (1991) 493–520. arXiv:<http://www.worldscientific.com/doi/pdf/10.1142/S0218127491000397>, doi:10.1142/S0218127491000397. URL <http://www.worldscientific.com/doi/abs/10.1142/S0218127491000397>
- [16] B. Krauskopf, H. M. Osinga, J. Galán-Vioque (Eds.), *Numerical continuation methods for dynamical systems, Understanding Complex Systems*, Springer, Dordrecht, 2007, path following and boundary value problems, Dedicated to Eusebius J. Doedel for his 60th birthday. doi:10.1007/978-1-4020-6356-5. URL <http://dx.doi.org/globalproxy.cvt.dk/10.1007/978-1-4020-6356-5>
- [17] D. Barkley, I. G. Kevrekidis, A. M. Stuart, The moment map: nonlinear dynamics of density evolution via a few moments, *SIAM J. Appl. Dyn. Syst.* 5 (3) (2006) 403–434. doi:10.1137/050638667. URL <http://dx.doi.org/globalproxy.cvt.dk/10.1137/050638667>
- [18] M. Elmegård, B. Krauskopf, H. Osinga, J. Starke, J. Thomsen, Bifurcation analysis of a smoothed model of a forced impacting beam and comparison with an experiment, *Nonlinear Dynamics* 77 (3) (2014) 951–966. doi:10.1007/s11071-014-1353-x. URL <http://dx.doi.org/10.1007/s11071-014-1353-x>
- [19] E. Bureau, F. Schilder, I. Santos, J. Thomsen, J. Starke, CONTINEX project home page at dtu, <http://www.continex.mek.dtu.dk/>.
- [20] L. Ljung (Ed.), *System Identification (2nd ed.): Theory for the User*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.
- [21] D. A. Barton, Control-based continuation: bifurcation and stability analysis for physical experiments, arXiv.org arXiv:1506.04052 [math.DS].